



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/829,499

04/22/2004

Cornell J. Kinderknecht

40003892-0056-002

6946

26263 7590 01/22/2009
SONNENSCHN NATH & ROSENTHAL LLP
P.O. BOX 061080
WACKER DRIVE STATION, SEARS TOWER
CHICAGO, IL 60606-1080

EXAMINER

VU, TUAN A

ART UNIT

PAPER NUMBER

2193

MAIL DATE

DELIVERY MODE

01/22/2009

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/829,499	Applicant(s) KINDERKNECHT ET AL.	
	Examiner TUAN A. VU	Art Unit 2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 12/19/08.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-33 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-33 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 12/19/08.

As indicated in Applicant's response, claims 1, 2, 5, 10, 19, 22 have been amended.

Claims 1-33 are pending in the office action.

Claim Objections

2. Claim 10 is objected to because of the following informalities: The phrase group (line 3) 'into the application process, the redirect code allows access to secured data ...' contains a grammatically incorrect independent verb-group (as underlined) because this verb-group clause has to subordinate to the independent main clause starting at 'A method for controlling ... comprising:'. A subordinate clause (including the verb 'allows') would have to be constructed within a scope of an explicit *conjunction of subordination*.
3. Claim 22 recites (line 8) 'one redirect function allows access to the at least ... computing system'; and this is also objected to because 'allows' is a grammatical impropriety as set forth above. Appropriate correction is strongly recommended.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5. Claims 1-21 rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention, in that these claims are rejected as being incomplete for omitting essential structural cooperative relationships of elements, such omission amounting to a gap between the necessary structural connections. See MPEP § 2172.01.

Art Unit: 2193

6. **As per claim 10**, The omitted structural cooperative relationships are: (i) pushing an injector, (ii) injecting a redirect code ... to allows access to secured data, (iii) executing the redirect code to reference redirect functions, (iv) resuming the execution of the application and (v) intercepting one target calls executing one redirect function in place of one target calls.

The steps of pushing in (i) is not related to the step (ii) of injecting a redirect code to access secured data. One cannot be ascertained as to what exactly the pushing as in (i) has to do with injecting of redirect code for the purpose of accessing secured data. Nor can one see how secured data is accessed by the injecting in (ii) in light of (iii) to reference a redirect library of redirect functions, or how redirect functions support the injecting step in (ii) regarding a remote computer and its security in place. There is no logical relationship or structural linkage between steps (i) (ii) and (iii). Further, there is no relationship between the referencing in step (iii) and the executing at least one redirect function as in (v) in terms of time relationship or code instantiation or availability, necessarily when step (iii) and step (v) does not clarify whether the referencing in (iii) allow any redirect function to persist until used in step (v). Further, there is no antecedent basis in the claim as a whole to clearly justify an actual stop/resume (of the application process) scenario so that step (iv) is plausible. The application process is re-executed as in the 'resuming' of (iv) cannot be meaningful when there is clear omission about it being stopped while executing in (iii) or injecting in (ii) or pushing in (i). The runtime availability of *redirect functions* from a library within the application process is not clearly established in the claim for one to construe how interception of target function calls would be dynamically supported by any redirect function, since executing of redirect code in (iii) does not lay out any time relationship between step (iii) or completion thereof, and step (iv) and/or the

Art Unit: 2193

beginning of (v) for one to see whether step (iii) has actually instantiated redirect code so that they become available after resuming in step (iv), and based on the indefinite existence of step (iv) it is hard to see how step (iii) interrelates to step (iv, resuming) and (v, intercepting and executing redirect functions). One cannot see whether the executing of 'redirect functions' --step v- is still within EITHER the process time of (iii) -- executing the redirect code in the application process -- OR the application process wherein step (iii) executes, notably when intervening step (iv) does not help establish a clear runtime continuity between (ii), (iii) and step (v).

Based on what appears to be a plurality of elements or steps as to interlink the elements of the claimed method as a whole, one cannot make use of the invention in order to achieve either step (ii), or step (v) or both. In view of the serious lack of structural (or functional) relationships between the steps as set forth above, step (ii) is treated as redirect code operating as a interjected call within an application and unrelated to step (i); such that the executing (of redirect code) in (iii) because of its intended use ('to reference') will be subsumed into the part of step (ii) and/or part of executing of step (v), and step (iv) will be given no patentable weight regarding 'resuming' because of its lack of antecedent basis.

Claims 11-21 are rejected for not remedying to the above indefinite language.

7. **As per claim 1**, there is omission of essential structural cooperative relationships of elements. These are linkage between (a) 'first computing device ... redirect code to allow access to secured data ... authorizing access with security in place at the remote computing' and (b) redirect code 'operable to intercept a set of target function calls ... execute the redirect functions ... target function calls'.

Art Unit: 2193

None of the recited library of redirect functions in regard to intercepting target functions calls during application process execution clearly establish structural or functional relationships between *intercepting* and the 'allows access to secured data at the remote computer system' limitation. One of ordinary skill in the art cannot discern or construe any time relationship between application process in execution including *interception of target function calls* as in (b) AND first computing device with memory having redirect code *for accessing* secured data as part of (a). Not a single relationship is conveyed from the interception of target calls of (b) to the first computing device memory, nor is there relationship between target function calls and secured data at the remote computer system, when 'target function calls' does not make it clear what 'target' is all about. Based on the indefinite nature of the qualifier 'target', it is hard to see how *target* functions inside an application process are related to secured data at a remote computing system as in (b). The secured data is not recited as having functional/structural link with a *library of* redirect functions and/or with target function calls, and one cannot learn whether target function calls actually implement access calls to obtain remote and secured data in a NW or a targeted device. When 'target' is not defined as having any relation to step (a) and that 'security in place' has no relationship to step (b), one of ordinary skill in the art would have to exert undue experimentation in order to achieve utilizing of some library of functions (referenced by a redirect code) via interception of target function calls (based on redirect code in memory as in a) thereby allowing secured data to be remotely accessed with security in place (emphasis added). Broader interpretation would have to be utilized to enable prosecution of the above un-related teachings or lack of cooperative relationship among step actions within the claimed method.

Claims 2-9 are also rejected for remedying to the lack of cooperative structural relationships.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

9. Claims 1-6, 10-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Calder et al, USPubN: 2002/0092003, and further in view Li et al, USPN: 7,406,533 (hereinafter Li)

As per claim 1, Calder discloses a system for controlling an application process comprising:

an injector operable stored on a computer readable medium (e.g. step 810, Fig. 8; para 0088, pg. 5 – Note: initializing package being transmitted – step 540 Fig. 5 – reads on injector code transmitted then stored in client application);

redirect code placed by the injector in a memory (e.g. interception module loaded ... application package starts - para 0088,pg. 5; step 540 Fig. 5; Fig. 9) operable to be placed in a memory of the application process; and

a library of redirect functions operable to be referenced by the redirect code (e.g. step 920 - Fig. 9; para 0096, pg. 5; step 620, interception module 810 -Fig. 8; para 0103, pg. 6) during the application process by the injector and

bypass security in place at a remote computing system (e.g. *approved network connection... participating client* – para 0088, pg. 5 -Note: 'bypass' given weight only as

Art Unit: 2193

scenario wherein an eligible user in a network receives accessible data – see USC 112, 1st paragraph; *transmitted ... to the client computer 140* – para 0092, pg. 5),

the redirect code operable to intercept a set of target function calls made by the application process (intercept 940 - Fig. 9; para 103, pg. 6) and execute the redirect functions for the intercepted target function calls (step 990 - Fig. 9).

Calder does not explicitly disclose redirect code for *allowing access to secured data at a remote computer system by authorizing access with security in place at the remote computing system*. Calder teaches using DLL library via an interception module and implementing redirection of these DLLs via wrapper routine implementation (see Calder: para 0103, pg. 6) as well as a proxy machine (see para 0121, pg. 8). Analogous to Calder's approach, **Li** discloses interception set up (like *tunneling module*) using interposed library (see *Lib 122b*, Fig. 3; Fig. 6A-B) to redirect calls destined to machines across established security machine or firewalls, including socket call interception for a given application data (see *client 120, Application 122a* - Fig. 3) ongoing at a client machine so that bi-directional data through each other side of firewall can be accessed (see *through firewall* – col. 4, lines 20-25; Fig. 3, 4A-B and related text). Based on Calder's teaching of proxy in place and secure keys to determine whether database content is sensitive (col. 7 para 0118-0119), of privileges approval of files targeted in Calder's interception module, and approval socket parameters (para 0132-0133, pg. 8-9; para 0155-0160, pg. 10) the security nature of connections between client and requested data is suggestive of access control of authorized content. Therefore, it would have been obvious for one skill in the art at the time the invention was made to implement the interception using DLLs in Calder so that a wrapping equivalent to that taught by Li would enable intercepted socket calls to be redirected to pass

Art Unit: 2193

through established firewalls as taught in Li, because this interception module would enable secured data to be promptly accessed directed for use at the requesting client's layer as desired by the client without extraneous involvement of the user or administrative agent, and whereby low-level security and/or protocol regulations across firewall would be still maintained or checked (see Li, col. 2 line 1 to col 3; Fig. 6B, Fig. 7, Fig. 9).

As per claims 2-4, Calder discloses wherein the injector is pushed to the first computing device (interception module - para 0096, pg. 5 Note: initializing package being transmitted to client – step 540 Fig. 5 – reads on injector code pushed onto client application) executing the application process; wherein the set of target function calls comprises socket function calls (e.g. Fig. 27); wherein the library of redirect functions comprises a dynamic link library (e.g. step 540-Fig.5; Fig. 9).

As per claim 5, Calder (in view of Li) discloses: a secure environment having a plurality of resources (e.g. *resource request 1335* - Fig. 13); a firewall securing all access to the plurality of resources in the secure environment (e.g. Fig. 22-24, 26; para 0076 - pg. 4; Fig. 39-40; page permissions 1325 -Fig. 13, Fig. 14-15); and policy identifying the resources authorized for access by the first computing device (*access LAN* - para 0074, 0076, pg. 3-4; Fig. 39-40; para 0131-0132 - Note: LAN, WAN and internal network based on access checking and encryption of data reads on policy to deny unauthorized intrusion).

Calder does not explicitly disclose access policy pushed to the first computing device identifying the resources authorized for access by the first computing device. Based on Calder electronic signing of transmitted package and decrypting at client end (para 0089, para 0092, pg. 5) and the rationale in claim 1 wherein proxy or firewalls can be in place as to filter or validate

Art Unit: 2193

socket content or algorithm to disallow non-approved request (see Calder: *socket information* – para 0152-0154 pg. 10; *pre-defined list* - para 0130-0132, pg. 8-9; see Li: Fig. 7-8; Fig. 21-23), it would have been obvious for one skill in the art at the time the invention was made to implement encrypted package in Calder so that they also contain meta-information regarding access policy (e.g. algorithm to uncode keys, or list of predefined keys, socket meta information, pre-defined list of approved connections) to access data in the scheme as intended by Calder to retrieve content or as in Li access of cross-firewall network data.

As per claim 6, Calder discloses wherein the application process comprises an application operable to communicate with the secure environment resources using an Internet transport protocol, the redirect code, and the redirect functions (e.g. Fig. 1-4; Fig. 9; para 103, pg. 6).

As per claim 10, Calder discloses a method for controlling an application process comprising:

pushing an injector to a first computing device executing the application process (refer to claim 2);

injecting a redirect code into the application process (refer to claim 1);

executing the redirect code in the application process to reference a redirect library (step 920 - Fig. 9; para 0096, pg. 5; refer to claim 1) of redirect functions;

resuming the execution of the application process (Note: “resuming” treated as execution of application in the course of referencing library and intercepting of socket calls – refer to USC 112 Rejection); and

intercepting at least one target function calls made by the application process and

Art Unit: 2193

executing at least one redirect function (step 990 - Fig. 9; refer to claim 1) in place of the at least one target function calls.

Calder does not explicitly disclose redirect code for *allowing access to secured data at a remote computer system by authorizing access with security in place at the remote computing system*. But this limitation has been addressed in claim 1.

As per claim 11, Calder discloses: starting the application process; interrupting the execution of the application process; and injecting the redirect code into a memory space of the application process (Fig. 10-11).

As per claim 12, Calder discloses wherein injecting a redirect code further comprises: starting the application process using a debug option; catching an exception thrown by the application process (see Fig 12-13 - Note: intercepting system call at low level reads on catching a exception at such level); locating memory space in the application process; injecting the redirect code into the memory space of the application process; and set an instruction pointer to the redirect code (e.g. step 1030 - Fig. 10; Fig. 15, 33, 41).

As per claim 13, Calder discloses wherein injecting a redirect code further comprises: starting the application process using a suspend option; creating memory space in the application process; injecting the redirect code into the memory space of the application process; and set an instruction pointer to the redirect code (e.g. Fig. 7, 10-11).

As per claim 14, Calder discloses wherein injecting a redirect code further comprises: starting the application process using a suspend option (see Fig. 12-13 Note: addressing a low level socket call with interception routine reads on a form of suspension enabling intercepting structure to be in place); creating memory space in the application process; injecting the redirect

Art Unit: 2193

code into the memory space of the application process; (Fig. 7, 10-11); and use a create remote thread function to execute the redirect code (e.g. Fig. 13; Fig. 15).

As per claim 15, Calder discloses wherein executing the redirect code comprises: loading the redirect library of redirect functions; determining a location of an import table replacement (Fig. 7,10 - Note: import table, export table reads on table of routines to insert to memory for replacement) function in the redirect library; and executing the import table replacement function.

As per claim 16, Calder discloses table including a dynamic link library (Fig. 10-11).

As per claim 17, Calder discloses wherein executing the import table replacement function comprises: searching an import table of the application process for the set of target function calls; and modifying the target function calls to reference redirect functions in the redirect library (Fig. 10-11).

As per claim 18 Calder discloses wherein executing the import table replacement function comprises: searching dynamic link libraries of the application process for the set of target function calls; and modifying the target function calls to reference redirect functions in the redirect library (e.g. para 0102, pg. 6).

As per claims 19-21, Calder discloses receiving user information; authenticating the user information; access policy specifying resources accessible by a user associated with the user information to a device used by the user; executing redirect functions to enable a secured access to a plurality of resources via a firewall (refer to claim 5)

Calder does not explicitly disclose access policy pushed to the first computing device identifying the resources authorized for access by the first computing device. But this has been

Art Unit: 2193

addressed in claim 5.

As per claim 20, refer to claim 3

As per claim 22, Calder discloses method comprising:

receiving user information; authenticating the user information (Fig. 18-19; re claim 5);
pushing an injector to a first computing device executing an application process (Fig. 8;
re claim 1); and

intercepting at least one target function call made by the application process to at least one of a plurality of secure resources (refer to claim 1) and executing at least one redirect function in place of the at least one target function call (step 540 Fig. 5; step 920 - Fig. 9; para 0096, pg. 5).

Calder does not explicitly disclose secure resources *at a remote computing system* and redirect function to *allow access to secured data at a remote computer system by authorizing access with security in place at the remote computing system*. However, this limitation has been addressed as obvious as set forth in claim 1.

As per claim 23, Calder discloses: injecting a redirect code into the application process; executing the redirect code in the application process to reference a redirect library of redirect functions; and resuming the execution of the application process (see claim 10).

As per claims 24-27, refer to claim 11-14, respectively.

As per claims 28-31, refer to claim 15-18, respectively

As per claims 32-33, refer to claim 20-21, respectively

Art Unit: 2193

10. Claims 7-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Calder et al, USPubN: 2002/0092003, and Li et al, USPN: 7,406,533; further in view of Thomas et al., USPN: 6,148,336 (hereinafter Thomas).

As per claim 7, Calder does not explicitly disclose wherein the application process comprises an email application. But GUI-based applications for which resources request are being fulfilled to support user's applications is disclosed (see Fig. 33-34; Fig. 47) in Calder's network of Lan users. Users applications having interception of messages with insertion of special code to redirect to a proper validating or readdressing of message request is disclosed in Thomas's Web-based paradigm (e.g. Fig. 6; *library ... containing a plug-in* - col. 9, lines 6-40) wherein socket communications are inserted with a plug-in supported via a DLL container for redirection with proper binding and re-wrapping (see Fig. 9-10). Based on Thomas' approach to introduce a novel way for addressing IP address filtering drawback wherein Email is one such application involving such filtering concern (see col. 2), it would have been obvious for one skill in the art to implement the application examination by Calder (see Fig. 33-34; *decrypt* - Fig. 39) so that the interception of LAN network messages via IP/TCP protocol via some dynamic application extension (such as plug-in as by Thomas -- see SUMMARY of Invention - col. 4-5) would be able examine the likes of Email message content and resolve potential incompatibility issues by this extension service such as examining, blocking, modifying, decrypting and re-encrypting prior to providing a wrap-up binding process (see Thomas, col. 5) which also endeavored as set forth above by Calder.

As per claims 8-9, Calder does not disclose wherein the application process comprises a web browser application wherein the application process comprises a file transfer application.

Art Unit: 2193

But applications with Winsock (see Thomas: see Fig. 1-6) or Windows system having provision of DLLs (see Calder para 0081-0082) was known environments in which standard file transfer and browser applications would have founded to provide communications between users and services, browser methodologies suggested as HTTP (see Li: HTTP tunneling – Fig. 4B) or pages in Calder: application page – Fig. 33). The limitation that applications be Email, or FTP or browser messages in light of the interception and redirection as taught by both Calder and Thomas would have been obvious for the same rationale as set forth above, because application like those require message transfer using a proper protocol, and the interception as purported by Calder or Thomas would support examination of such message internals to provide a modified and adjusted redirection as mentioned above in the respective endeavor by Calder and Thomas.

Response to Arguments

11. Applicant's arguments filed 10/19/08 have been fully considered but they are moot in light of the new grounds of rejection which have been necessitated by the Amendments.

Conclusion

12. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (571) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on (571)272-3759.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before

Art Unit: 2193

using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Tuan A Vu/

Primary Examiner, Art Unit 2193

January 16, 2009